ECE 375 Prelab for lab 8

Remotely Operated Vehicle (USART)

Lab Time: Friday 4 - 6

Faaiq Waqar

PreLab Questions

To complete this prelab, you may find it useful to look at the full ATmega128 datasheet. If you consult any online sources to help answer the pre lab questions, you must list them as references in your prelab.

1. In this lab, you will be given a set of behaviors/actions that you need to have a proof-of-concept "toy" perform. Think of a toy you know of (or look around online for a toy) that is likely implemented using a microcontroller, and **describe the behaviors it performs**. Here is an example of behavior: "If you press button X on the toy, it takes action Y (or makes sound Z)".

Toy: Tickle Me Elmo

Behaviour 1: If you press the button on the center of the toy, the toy will emit a giggling sound, and say one of two phrases

Behaviour 2: If you press the button on the center of the toy three times in a row, the toy will emit a laughing sound, and the toy will begin to shake, emulating large amounts of laughter

2. For each behavior you described in the previous question, explain which microcontroller feature was likely used to implement that behavior, and give a brief code example indicating how that feature should be configured. Make your explanation as **ATmega128-specific as possible** (e.g., discuss which I/O registers would need to be configured, and if any interrupts will be used), and also mention if any additional mechanical and/or electronic devices are needed.

Behaviour 1: For both of these, I will work under the assumption that we will use an interrupt that will work so long as the toy is on. We will set an interrupt at the top, perhaps INT0 just for namesake, setting the routine location to $0002, which will jump to a subroutine that will need to use an I/O speaker to transmit this sound. Now, admittedly, I do not know how to transmit sound, so I will call that pseudocode in the program example.

Behaviour 2: For this instance, we can use the same interrupt subroutine used in the behaviour 1. However, in this instance, we will want to implement a counter so that we know when we have reached 3 button pushes. This can be simply done by setting one of our registers to be set specifically to counting. Now, for the actual movement outside of the laughing, we can implement a fast pwm adjustment that will go from motor stop to a motor drive for a set amount of time.

Example Program:

```
.org $0000

        rjmp    INIT

.org $0002

        rjmp    TickleMe

        reti

.org $0046
```

```
INIT:
        .def counter = r17
        .def mpr = r16
        ldi mpr, high(RAMEND)
        out SPH, mpr
        ldi mpr, low(RAMEND)
        out SPL, mpr
        ldi mpr, (1<<WGM01|1<<WGM00|1<<COM01|1<<CS00)
        out TCCR0, mpr
        ldi             mpr, 0b00000001
        out             DDRB, mpr
        ldi             mpr, 0b11111110
        out             PORTB, mpr

MAIN:
        rjmp MAIN
TickleMe:
        inc     counter
        cpi     counter, 3
        breq    TickleExt
        // Speaker Output Single / Double Press
        ret
TickleExt:
        ldi     counter, 0
        //Speaker Output
        ldi             mpr, $FF        ; Loaded maximum value into mpr
        out             OCR0, mpr           ; Output into Output compares

        ret
```

3. Each ATmega128 USART module has two flags used to indicate its current **transmitter** state: the Data Register Empty (UDRE) flag and Transmit Complete (TXC) flag. What is the difference between these two flags, and which one always gets set first as the transmitter runs? You will probably need to read about the Data Transmission process in the datasheet (including looking at any relevant USART diagrams) to answer this question.

TXC: Transmit completion is set when the entire frame in the Transmit shift register has been moved out and there is no new data in the transmission buffer. The flag will be cleared when the transmission completion interrupt is executed.

UDRE: Indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and will be cleared with the buffer contents new data to be transmitted that has yet to be shifted into the shift register

UDRE will be set first in program, as the TXC is reliant on transmission before setting

4. Each ATmega128 USART module has one flag used to indicate its current receiver state (not including the error flags). **For USART1 specifically**, what is the name of this flag, and what is the interrupt vector address for the interrupt associated with this flag? This time, you will probably need to read about Data Reception in the datasheet to answer this question.

RXC or receive complete flag indicates that there is unsafe data present in the buffer. Associated with the interrupt vector address $003C